

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

CONTENT-BASED ROUTING

Inventor(s):
Saurab Nog
Alfred Lee
David Levin

ATTORNEY'S DOCKET NO. MS1-1854US
SENDING NODE'S DOCKET NO. 306880.01

CONTENT-BASED ROUTING

TECHNICAL FIELD

[0001] The described subject matter relates to electronic computing, and more particularly to systems and methods of content-based routing in electronic computing networks.

BACKGROUND

[0002] In large networks such as the Internet, data packets are routed to a destination by routers. These routers send and receive messages but do not otherwise process the message. The routers typically do not even maintain any information about the data packet much less communicate other information about the data packet to other routers or to the sender.

[0003] Overlay networks may be provided on the existing network infrastructure to process data packets at the application level, and therefore are able to provide additional routing services (e.g., security). Overlay networks are implemented by overlay nodes linked to one another by overlay links. Each overlay link may include many physical links in the underlying network infrastructure.

[0004] Overlay nodes process every data packet at the application level before resending the data packet to the next node in the overlay network. In addition, overlay nodes may not be optimally positioned because there is often

1 little or no knowledge of the physical layout of the underlying network
2 infrastructure. These factors contribute to longer latency in the overlay network. In
3 addition, overlay routers may become a central point of failure. For example, if
4 particular types of messages have to be routed by a single overlay router and that
5 router fails, the messages do not reach their intended destination.
6

7 SUMMARY

8 [0005] Implementations are described and claimed herein for content-based
9 routing in an overlay network. According to an exemplary implementation, data
10 packets are received by routing nodes in an overlay network. In addition to
11 forwarding the message to another routing node or to the destination node, the
12 routing nodes also return routing policies to the sending node.
13

14 [0006] The routing policies include instructions for redirecting messages
15 based on content of the message. For example, when a message is received by a
16 first routing node and redirected to a second routing node, the first routing node
17 may also return a routing policy to the sending node. The routing policy instructs
18 the sending node to bypass the first routing node and issue other messages with
19 similar content directly to the second routing node.
20

21 [0007] Before issuing a message, the sending node determines which
22 policies apply to the message, and may also apply other policies (e.g., security
23 policies) to the message. The sending node iterates through the routing policies,
24
25

1 modifying the address in the message according to the instructions included in
2 applicable routing policies. For example, the message may be modified based on a
3 routing policy so that instead of sending the message to a first routing node, the
4 message is sent to a second routing node. The message may be modified again
5 based on another routing policy so that instead of sending the message to the
6 second routing node, the message is sent directly to a destination node. The
7 sending node is able to bypass one or more intermediary nodes (or routing nodes),
8 reducing latency in the overlay network and reducing or eliminating the occurrence
9 of central failures. In addition, the sending node is able to automatically discover
10 and refine shortcuts in the overlay network.
11

12 [0008] In some implementations, articles of manufacture are provided as
13 computer program products. One implementation of a computer program product
14 provides a computer program storage medium readable by a computer system and
15 encoding a computer program for content-based routing. Another implementation
16 of a computer program product may be provided in a computer data signal
17 embodied in a carrier wave by a computing system and encoding the computer
18 program for content-based routing.
19

20 [0009] The computer program product encodes a computer program for
21 executing a computer process on a computer system that includes receiving a
22 message at a routing node in an overlay network, and generating a routing policy
23 for a sending node based at least in part on the message content.
24
25

1 [0010] In another implementation of the computer program product, a
2 computer process includes identifying at least one routing policy for a message
3 based on the message content, and changing an address in the message to bypass at
4 least one node in an overlay network based on the at least one routing policy.

5 [0011] In yet another implementation, a method is provided. The method
6 includes receiving a message at a routing node in an overlay network, and
7 generating a routing policy for a sending node based at least in part on the message
8 content.

9 [0012] In another implementation, a method includes identifying at least one
10 routing policy for a message based on the message content, and changing an
11 address in the message to bypass at least one node in an overlay network based on
12 the at least one routing policy.

13 [0013] In yet another implementation, a system is provided including a
14 routing node receiving a message in an overlay network. A message processor is
15 provided at the routing node. The message processor generates a routing policy for
16 a sending node of the message based at least in part on the message content.

17 [0014] In another implementation of the system a messaging module at the
18 sending node changes an address in a message to bypass at least one node in an
19 overlay network based on at least one routing policy.
20
21
22
23
24
25

BRIEF DESCRIPTION OF THE DRAWINGS

1 [0015] Fig. 1 is a schematic illustration of an exemplary computer network
2 that may implement content-based routing;
3

4 [0016] Fig. 2 is a schematic diagram of exemplary nodes in an overlay
5 network that may implement content-based routing;

6 [0017] Fig. 3 is a high-level diagram illustrating an exemplary
7 implementation of content-based routing in an overlay network;

8 [0018] Fig. 4 is another high-level diagram illustrating an exemplary
9 implementation of content-based routing in an overlay network;
10

11 [0019] Figs. 5-6 are flowcharts illustrating exemplary operations to
12 implement content-based routing; and

13 [0020] Fig. 7 is a schematic illustration of an exemplary computing device
14 that can be utilized to implement content-based routing.
15

DETAILED DESCRIPTION

16
17
18 [0021] Briefly, routing nodes may be implemented in an overlay network to
19 return routing policies to a sending node. The routing policies include instructions
20 to bypass one or more intermediary nodes in the overlay network. The sending
21 node may apply these routing policies to data packets (or messages) to reduce
22 latency in the overlay network.
23
24
25

1 [0022] For purposes of illustration, a user attempting to purchase a book
2 from a commercial web service (e.g., msn.com) may issue a message for the
3 commercial web service which is received by a routing node in the overlay
4 network. The routing node resends the message, e.g., to a destination node in the
5 bookstore division at the commercial web service. In addition to forwarding the
6 message, the routing node also returns a routing policy to the sending node.

7 [0023] The routing policy includes instructions for sending similar
8 messages. In this example, an overlay router or routing node may return a routing
9 policy that instructs the sending node to issue book purchasing messages directly
10 to the node for the bookstore division at the commercial web service.
11

12 [0024] The sending node maintains these routing policies (e.g., along with
13 other policies) to apply to new messages. Before sending a new message, the
14 sending node iterates through the routing policies, modifying the address in the
15 message according to applicable routing policies. In this example, the message
16 may be modified based on the routing policy so that the message is sent directly to
17 the destination node and bypasses one or more intermediary or routing nodes in the
18 overlay network.
19
20

21 **Exemplary System**

22 [0025] Fig. 1 is a schematic illustration of an exemplary computer network
23 that may implement content-based routing. The networked computer system 100
24
25

1 may include one or more communication networks, such as local area network
2 (LAN) 110 and/or wide area network (WAN) 120. One or more sending devices
3 140a-d may be linked to one or more destination nodes 150a-f over the
4 communication network(s) 110, 120.

5 [0026] Sending devices 140a-d and destination nodes 150a-f connect to a
6 network via a communication connection such as, e.g., an Ethernet connection.
7 Although there are no theoretical limits on the number of computing devices that
8 can be included in a network such as network 100, the number of computing
9 devices are limited primarily by the connectivity implemented in the network.
10

11 [0027] The term "node" is used herein to refer to sending nodes, routing
12 nodes, and destination nodes and includes the hardware and software (i.e., the
13 entire computer system) used to perform various computing operations. For
14 purposes of illustration, a sending node may be implemented, e.g., as a stand-alone
15 personal desktop or laptop computer (PC), workstation, personal digital assistant
16 (PDA), or electronic appliances, to name only a few examples. A destination node
17 may be implemented, e.g., as a server computer that is dedicated to server
18 applications or that also runs other applications. Other implementations are also
19 contemplated and are not limited to the examples described above.
20

21 [0028] It is noted that a destination node may be operatively associated with
22 one or more resources. Resources available via the destination node may include
23 other computing or data processing systems, storage, or other devices. The
24
25

1 destination node may also provide the sending node with services, such as
2 transaction processing, interactive web pages, email, etc.

3 [0029] An overlay network 180 may be implemented, e.g., on the network
4 infrastructure shown in Fig. 1. Overlay network 180 includes overlay nodes, such
5 as, e.g., routing node 160 and destination nodes 170a, 170b. Overlay nodes may be
6 implemented as server computers or other computing devices. In Fig. 1 for
7 example, routing node 160 is implemented at server computer 150b and
8 destination nodes 170a, 170b are implemented at server computers 150e, 150f
9 which provide access to services 175a, 175b (e.g., web applications).
10

11 [0030] Messages are transmitted between overlay nodes via one or more
12 overlay links (not shown). Overlay links may include many physical links (e.g.,
13 router devices and server computers) in the underlying network(s) 110, 120.
14 Routing node(s) 160 receive and reissue messages in the overlay network 180
15 based on routing table(s) 165.
16

17 [0031] Routing node(s) 160 process messages in the overlay network 180 at
18 the application level and may be implemented to return one or more routing
19 policies 190 to, e.g., one or more of the sending nodes. The routing policies 190
20 include instructions that the sending node may apply to similar messages to bypass
21 one or more of the routing nodes 160, as will be discussed in more detail below.
22

23 [0032] **Fig. 2** is a schematic diagram of exemplary nodes that may
24 implement content-based routing in an overlay network. Sending node 200 may
25

1 connect to a destination node 210 via an overlay network. In Fig. 2, sending node
2 200 connects to the destination node 210 via port 201 and data packets (or
3 messages) issued by the sending node 200 may be delivered to the destination node
4 210 via one or more routing nodes 220a, 220b. Alternatively, messages may bypass
5 routing nodes 220a, 220b and may be delivered directly to destination node 210, as
6 illustrated by dashed line 225 and discussed in more detail below.

7 [0033] Referring to Fig. 2, sending node 200 may include a messaging
8 module 230 operatively associated with a policy cache 240. Messaging module
9 230 may be implemented in computer-readable program code (e.g., software
10 and/or firmware) stored in computer-readable storage or memory and executable
11 by a processor (or processing units) at the sending node 200. Messaging module
12 230 may include a message generator 232, message processing module 234, and
13 policy manager 236.
14

15 [0034] Message generator 232 may be implemented to generate data packets
16 (or messages) to communicate with the destination node 210. For example, a
17 message may include a request to access a service (e.g., a web application or
18 remote database) via the destination node 210. Of course messages are not limited
19 to such requests and may also include, by way of example, credit card information,
20 security credentials, and data for remote storage, to name only a few such
21 examples.
22
23
24
25

1 [0035] Message processing module 234 may be implemented to prepare
2 messages to be issued in an overlay network. For example, the message processing
3 module 234 may cooperate with policy manager 236 to apply one or more policies
4 to the message before it is issued.

5 [0036] Policy manager 236 is operatively associated with policy cache 240.
6 Policy cache 240 may be stored in memory. Policy cache 240 may include one or
7 more policies, such as, e.g., routing policy 242, transport policy 244, and any
8 number of other types of policies 246 (e.g., a security policy).
9

10 [0037] In an exemplary implementation, the Web Services Policy (WS-
11 Policy) specification defines a general model and syntax for policy expressions.
12 The policy may be expressed in machine-readable extensible markup language
13 (XML) format to facilitate interoperability between different platforms and web
14 services infrastructure. Of course a policy is not limited to any particular syntax or
15 format and other implementations are also possible.
16

17 [0038] Routing policies 242 may be received at the sending node 200 from
18 one or more routing nodes 220a, 220b, e.g., in response to issuing messages as
19 discussed in more detail below. Routing policies 242 identify direct paths to other
20 nodes that bypass one or more routing nodes 220a, 220b in the overlay network.
21

22 [0039] Transport policies 244 identify the preferred, or in some cases, the
23 required transport protocol to communicate with a node in the overlay network.
24 For purposes of illustration, the transport protocol may be Simple Object Access
25

1 Protocol (SOAP). SOAP is a messaging protocol used to encode transactions for
2 transfer over a network using any of a variety of Internet protocols (e.g., HTTP,
3 SMTP, MIME). SOAP messages do not need to be formatted for use with any
4 particular operating system, making SOAP messages commonplace in network
5 environments. However, other transport protocols are also contemplated and may
6 include, e.g., HTTP, SMTP and other protocols now known or that may be later
7 developed.

8 [0040] In an exemplary implementation, policies are applied to messages in
9 the following order: 1) security policies, 2) routing policies, 3) transport policies,
10 and 4) general policies such as, e.g., compression and encryption policies.
11 Transport policies are applied after the routing policies because the transport
12 protocol may depend at least to some extent on the message address, and may
13 change as the routing policies are applied to the message. It is noted, however, that
14 in other implementations the policies do not have to be applied in any particular
15 order.
16

17 [0041] It is noted that the sending node 200 is not limited to the exemplary
18 implementation shown in Fig. 2. For example, the functions do not need to be
19 embodied in separate modules. In yet other implementations, additional functional
20 components may also be included.
21

22 [0042] Continuing our discussion with reference to Fig. 2, routing nodes
23 220a, 220b (hereinafter generally referred to as reference nodes 220) may be
24
25

1 implemented in the overlay network to receive messages, process messages, and
2 resend the messages to another node in the overlay network (e.g., to another
3 routing node or a destination node). Routing nodes 220 pass the message to the
4 application level for processing (e.g., at message processor 250a, 250b). Message
5 processor 250a, 250b may be implemented as computer-readable program code
6 (e.g., software and/or firmware) stored in computer-readable storage or memory
7 and executable by a processor (or processing units) at the routing node 220.

8 [0043] Message processors 250 are operatively associated with routing
9 tables 260a, 260b (hereinafter generally referred to as routing tables 260). Routing
10 tables 260 identify one or more paths to route messages in the overlay network.
11 Routing tables 260 may identify other routing nodes or a destination node based
12 on, e.g., the type of message or message content. Message processors 250 use the
13 routing tables 260 to determine an address to reroute a message in the overlay
14 network.
15

16 [0044] For purposes of illustration, a message including a request to
17 purchase a book may be issued to a routing node that processes book purchasing
18 requests. The routing node may process the message and determine that the request
19 is for a fiction novel. The message is then reissued to a destination node that
20 processes requests for fiction novels. Alternatively, if the routing node determines
21 that the request is for a technical textbook, the request is routed to a destination
22 node that processes requests for technical textbooks.
23
24
25

1 [0045] **Fig. 3** is a high-level diagram illustrating an exemplary
2 implementation of content-based routing in an overlay network. In this illustration,
3 sending node 300 generates and issues message 310 including a request to
4 purchase a book from a book selling service (e.g., Service A) at a commercial web
5 site (e.g., one of the destination nodes 330a).

6 [0046] Message 310 may include a header 312 identifying an address for the
7 message 310, and a body 314, e.g., containing a request to purchase the book. In an
8 exemplary implementation, the address is identified in the header 312 of a SOAP
9 message 310 using XPath expressions.
10

11 [0047] XPath provides common syntax and semantics for XSL
12 Transformations (XSLT) and XPointer. XPath uses path notation to navigate the
13 XML document hierarchy and operates on the logical structure of an XML
14 document instead of its surface syntax, allowing manipulation of strings, numbers
15 and Boolean expressions in XML documents. However, implementations are not
16 limited to XPath expressions.
17

18 [0048] Although the sending node 300 may not have an address for the
19 destination node, the sending node 300 may have an address for a neighboring
20 node (e.g., routing node 340a) that can process and reroute the message 310.
21 Accordingly, sending node 300 issues message 310 to routing node 340a. The
22 message 310 is passed to the application level (e.g., message processor 250 in Fig.
23 2) for processing at the routing node 340a. Processing may include determining a
24
25

1 type of message based on the message content. The message content may include
2 information included in header 302 and/or message body 304. Routing node 340a
3 accesses a routing table 350a to determine an address to route the message 310 to
4 based at least in part on the message content. An exemplary routing table that may
5 be implemented at routing node 340a is illustrated in Table 1.
6

7
8 **Table 1: Exemplary Routing Table 1**

9

Message Type	Address
Purchase Book	Routing Node 2
Open Account	Routing Node n

10
11
12

13 [0049] Exemplary Routing Table 1 illustrated above by Table 1 includes
14 categories or types of messages and a corresponding address for the message and
15 may be used to identify another address for the message 310 based on the message
16 content. As an illustration, if message 310 includes a request to purchase a book,
17 the message address is modified to "Routing Node 2", e.g., by modifying message
18 header 302 at the application level at routing node 340a. Alternatively, if the
19 message 310 includes a request to open a new account, the address is modified to
20 "Routing Node n."
21

22 [0050] After modifying the address in message 310, the message is passed
23 to the messaging level where it is reissued by routing node 340a in the overlay
24
25

network, e.g., to routing node 340b (Routing Node 2) or routing node 340c (Routing Node n).

[0051] When the message 310 is received by the next routing node (e.g., routing node 340b), the message 310 is again passed to the application level to determine the message content and another address for the message, e.g., based on routing table 350b. An exemplary routing table 350b that may be implemented at routing node 340b is illustrated in Table 2.

Table 2: Exemplary Routing Table 2

Message Type	Address
Purchase Request	
*Fiction	Destination Node A
*Textbook	Destination Node B

[0052] Again, exemplary Routing Table 2 illustrated above in Table 2 includes categories or types of messages and a corresponding address which may be used to identify the next node for message 310 based on the message content. As an illustration, if message 310 includes a request to purchase a fiction novel, the address of the message is modified to "Destination Node A", e.g., by modifying the message header 302 at the application level at routing node 340b. Alternatively, if the message 310 includes a request to purchase a textbook, the address of the message is modified to "Destination Node B."

1 [0053] In addition to processing and reissuing message 310, one or more of
2 the routing nodes 340a, 340b may also return a routing policy 360a, 360b to the
3 sending node 300. As discussed above, routing policies include instructions for
4 routing messages based on the message content to bypass one or more of the
5 routing nodes in an overlay network. Routing policies 360a, 360b are maintained
6 by the sending node 300, e.g., in cache 305.

7 [0054] It is noted that more routing policies received by a sending node tend
8 to refine the granularity of content-based routing in the overlay network. For
9 example, if a single routing node returns a routing policy to the sending node, new
10 messages may be issued by the sending node based on the routing policy to bypass
11 the single routing node. However, if multiple routing nodes return a plurality of
12 routing policies, new messages may be issued by the sending node based on these
13 routing policies to bypass a plurality of routing nodes. For example, if only routing
14 policy 360a is received, the sending node can shortcut Routing Node 2. But if both
15 routing policies 360a, 360b are received, the sender can bypass both intermediate
16 nodes and be issued directly to the destination node. In addition, sending messages
17 to a plurality of destination nodes tends to increase the number of routing policies
18 that are received by the sending node, increasing the number of routing policies
19 that the sending node has available for different types of messages.
20
21

22 [0055] Still other implementations are also contemplated. For example,
23 although the routing policies have been illustrated as being returned to the sending
24
25

1 node in Fig. 3, the routing policies may be returned to other nodes in the overlay
2 network (e.g., one or more of the routing nodes). These other nodes may then
3 apply the routing policies to messages to bypass other intermediary nodes in the
4 overlay network. For example, if routing nodes 340a, 340b exchanged a routing
5 policy regularly, then the first message issued by the sender might result in routing
6 node 340a returning both routing policies 360a, 360b and also allow routing node
7 340a to send a message directly to the destination node.

8 [0056] In another exemplary implementation, the routing nodes may process
9 the message to include a routing policy within the message 310 itself (e.g., as part
10 of the message header 312 or message body 314). When the message 310 reaches
11 an endpoint (e.g., a destination node 320), the node extracts the routing policies
12 from the message 310 and returns one or more of the routing policies to the
13 sending node.
14

15 [0057] In yet another exemplary implementation, the routing policies do not
16 need to be returned to the sending node if the sending node has already received a
17 routing policy. In still another exemplary implementation, the routing policies may
18 be combined into a master routing policy.
19

20 [0058] **Fig. 4** is another high-level diagram illustrating an exemplary
21 implementation of content-based routing in an overlay network. In this illustration,
22 a sending node 400 issues a message including a request to purchase a book from a
23 book selling service 420 at destination node 430 (e.g., a commercial website).
24
25

[0059] The sending node 400 generates a message 410a which may include a header 402a including a node address, and a body 404, e.g., having the book purchase request. Before issuing the message 410a, sending node 400 determines if one or more routing policies (e.g., maintained in cache 405) apply to the message 410a.

[0060] In an exemplary implementation, routing policies were already received from routing nodes 440a, 440b in response to sending similar messages, as described above with reference to Fig. 3. The sending node 400 may access one or more of these routing policies from cache 405 and determine whether one or more of the routing policies apply to the message 410a.

[0061] Exemplary routing policies are illustrated in Tables 3 and 4.

Table 3: Exemplary Routing Policy 1

Condition	Instruction
IF Operation = Purchase Book AND Address = Routing Node 1	THEN Change Address to Routing Node 2

Table 4: Exemplary Routing Policy 2

Condition	Instruction
IF BookID = Fiction Novel AND Address = Routing Node 2	THEN Change Address to Destination node A

[0062] The exemplary routing policies illustrated above in Tables 3 and 4 include one or more conditions and corresponding instruction(s) to be implemented if the condition(s) are satisfied. XPath expressions may be used in an exemplary implementation to define the conditions. XPath expressions may be readily tested to determine if a message matches a pattern (e.g., if the message content is similar to message content identified by a routing policy). Other implementations are also contemplated, however, and are not limited to XPath expressions. In any event, the routing policies may be used by the sending node to modify the message address.

[0063] In operation, the sending node 400 may iterate through the routing policies and apply routing policies based on the message content. For purposes of illustration, if the message 410 contains a request to purchase a book and the address is "Routing Node 1," then the address of the message 410a may be modified to "Routing Node 2" based on the routing policy illustrated in Table 3. Furthermore, if the modified message 410b contains a request to purchase a fiction novel and the address is "Routing Node 2," then the address of the message 410b

1 is again modified, this time from "Routing Node 2" to "Destination Node A" based
2 on the routing policy illustrated in Table 4.

3 [0064] In an exemplary implementation, the routing policies are applied to
4 the message 410a and then to the modified message 410b in an ordered manner.
5 That is, the routing policy identifying the first routing node is applied before the
6 routing policy identifying the second routing node so that each of the conditions
7 are satisfied and each of the routing policies is properly applied. However,
8 applying the routing policies to a message is not limited to any particular order.
9

10 [0065] After iterating through the routing policies, sending node 400 may
11 issue the modified message 410c in the overlay network. The modified message
12 410c is sent directly to Destination Node A (node 430), bypassing routing nodes
13 440a, 440b.

14 [0066] It is noted that although message 410c is illustrated in Fig. 4 as
15 bypassing each intermediary node 440a and 440b, messages do not need to always
16 be modified for direct routing to a destination node. For example, a message may
17 be issued by the sending node directly to "Routing Node 2," only bypassing
18 "Routing Node 1" in the example illustrated in Fig. 4.
19
20

21 Exemplary Operations

22 [0067] Described herein are exemplary methods for implementing content-
23 based routing in a network environment, such as the exemplary overlay networks
24
25

described above. The methods described herein may be embodied as logic instructions on one or more computer-readable medium. When executed on a processor, the logic instructions cause a general purpose computing device to be programmed as a special-purpose machine that implements the described methods. In the following exemplary operations, the components and connections depicted in the figures may be used to implement content-based routing in a computer network.

[0068] **Fig. 5** is a flowchart illustrating exemplary operations 500 that may implement content-based routing. The operations may be implemented, e.g., at a routing node, to return routing policies to a sending node for bypassing one or more intermediary nodes in an overlay network when the sending node issues similar messages.

[0069] In operation 510, a routing node receives a message that was issued by a sending node in an overlay network. In operation 520, the message is passed to the application level for processing by the routing node. In operation 530, the routing node generates a routing policy based on the message. In operation 540, the routing node returns the routing policy to the sending node. The message is forwarded in operation 550. The routing node may forward the message to another routing node, or alternatively, to a destination node.

[0070] **Fig. 6** is another flowchart illustrating exemplary operations 600 that may be implemented for content-based routing. The operations may be

1 implemented, e.g., at a sending node, to apply one or more routing policies to a
2 message. The message may then be issued to a destination node, bypassing one or
3 more intermediary nodes in an overlay network.

4 [0071] In operation 610 a sending node generates a message. The sending
5 node determines an address to send the message in operation 620. The sending
6 node may also identify one or more policies for the message. These policies may
7 include but are not limited to, e.g., data compression, encryption, and security
8 parameters to apply to the message. The sending node may also identify a routing
9 policy in operation 630.
10

11 [0072] In operation 640, the routing policy is applied to the message. For
12 example, the message may be modified so that it is sent directly to another routing
13 node or to the destination node. In operation 650 the sending node iterates through
14 operations 630-640 until each of the identified routing policies have been applied
15 to the message. In operation 660, the sending node issues the message in the
16 overlay network.
17

18 [0073] It is noted that the operations shown in Fig. 5 and Fig. 6 are merely
19 illustrative of exemplary implementations of content-based routing and are not
20 intended to limit the scope of the invention to any particular operations or order.
21
22
23
24
25

Exemplary Computing Device

1 [0074] Fig. 7 is a schematic illustration of an exemplary computing device
2
3 700 that can be utilized to implement a node in an overlay network. Computing
4 device 700 includes one or more processors or processing units 732, a system
5 memory 734, and a bus 736 that couples various system components including the
6 system memory 734 to processors 732. The bus 736 represents one or more of any
7 of several types of bus structures, including a memory bus or memory controller, a
8 peripheral bus, an accelerated graphics port, and a processor or local bus using any
9 of a variety of bus architectures. The system memory 734 includes read only
10 memory (ROM) 738 and random access memory (RAM) 740. A basic input/output
11 system (BIOS) 742, containing the basic routines that help to transfer information
12 between elements within computing device 700, such as during start-up, is stored
13 in ROM 738.
14

15 [0075] Computing device 700 further includes a hard disk drive 744 for
16 reading from and writing to a hard disk (not shown), and may include a magnetic
17 disk drive 746 for reading from and writing to a removable magnetic disk 748, and
18 an optical disk drive 750 for reading from or writing to a removable optical disk
19 752 such as a CD ROM or other optical media. The hard disk drive 744, magnetic
20 disk drive 746, and optical disk drive 750 are connected to the bus 736 by
21 appropriate interfaces 754a, 754b, and 754c. The drives and their associated
22 computer-readable media provide nonvolatile storage of computer-readable
23
24
25

1 instructions, data structures, program modules and other data for computing device
2 700. Although the exemplary environment described herein employs a hard disk, a
3 removable magnetic disk 748 and a removable optical disk 752, other types of
4 computer-readable media such as magnetic cassettes, flash memory cards, digital
5 video disks, random access memories (RAMs), read only memories (ROMs), and
6 the like, may also be used in the exemplary operating environment.

7 [0076] A number of program modules may be stored on the hard disk 744,
8 magnetic disk 748, optical disk 752, ROM 738, or RAM 740, including an
9 operating system 758, one or more application programs 760, other program
10 modules 762, and program data 764. A user may enter commands and information
11 into computing device 700 through input devices such as a keyboard 766 and a
12 pointing device 768. Other input devices (not shown) may include a microphone,
13 joystick, game pad, satellite dish, scanner, or the like. These and other input
14 devices are connected to the processing unit 732 through an interface 756 that is
15 coupled to the bus 736. A monitor 772 or other type of display device is also
16 connected to the bus 736 via an interface, such as a video adapter 774.
17
18

19 [0077] Generally, the data processors of computing device 700 are
20 programmed by means of instructions stored at different times in the various
21 computer-readable storage media of the computer. Programs and operating systems
22 may be distributed, for example, on floppy disks, CD-ROMs, or electronically, and
23 are installed or loaded into the secondary memory of a computer. At execution, the
24
25

1 programs are loaded at least partially into the computer's primary electronic
2 memory.

3 [0078] Computing device 700 may operate in a networked environment
4 using logical connections to one or more remote computers, such as a remote
5 computer 776. The remote computer 776 may be a personal computer, a server, a
6 router, a network PC, a peer device or other common network node, and typically
7 includes many or all of the elements described above relative to computing device
8 700. The logical connections depicted in Fig. 7 include a LAN 780 and a WAN
9 782.
10

11 [0079] When used in a LAN networking environment, computing device
12 700 is connected to the local network 780 through a network interface or adapter
13 784. When used in a WAN networking environment, computing device 700
14 typically includes a modem 786 or other means for establishing communications
15 over the wide area network 782, such as the Internet. The modem 786, which may
16 be internal or external, is connected to the bus 736 via a serial port interface 756.
17 In a networked environment, program modules depicted relative to the computing
18 device 700, or portions thereof, may be stored in the remote memory storage
19 device. It will be appreciated that the network connections shown are exemplary
20 and other means of establishing a communications link between the computers
21 may be used.
22
23
24
25

1 [0080] Nodes may include adapter hardware and software to enable a
2 connection to the communication network. The connection to communication
3 network may be through an optical coupling or more conventional conductive
4 cabling depending on the bandwidth requirements. An adapter may be
5 implemented as a plug-in card on computing device 700. Nodes may implement
6 any number of adapters to provide as many connections to communication network
7 as the hardware and software support.

8 [0081] In addition to the specific implementations explicitly set forth herein,
9 other aspects and implementations will be apparent to those skilled in the art from
10 consideration of the specification disclosed herein. It is intended that the
11 specification and illustrated implementations be considered as examples only, with
12 a true scope and spirit of the following claims.
13
14
15
16
17
18
19
20
21
22
23
24
25